

**IN THE CLAIMS:**

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~striketrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please AMEND claims 12, 15 and 18-20 and ADD claim 24 in accordance with the following:

1-10. (CANCELLED)

11. (PREVIOUSLY PRESENTED) The arithmetic and logic unit as claimed in claim 12, wherein the fourth part performs initialization based on prediction information given to the branch instruction.

12. (CURRENTLY AMENDED) An arithmetic and logic unit, comprising:  
a first part performing a branch prediction in response to a branch instruction;  
a second part updating a transition probability of the branch prediction according to whether a branch is actually made;  
a third part detecting that a process is switched; and  
a fourth part initializing branch prediction information when the third part detects that the process is switched, and  
wherein the fourth part fixedly performs initialization of the branch prediction information by determining an initialization value according to a comparison of a program counter value with a branch destination address and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN) ~~of the branch instruction~~ to set the branch prediction information to predetermined branch prediction information regardless of past branch prediction results, without depending on a particular process.

13. (CANCELLED)

14. (PREVIOUSLY PRESENTED) The method as claimed in claim 15, wherein said initializing comprises performing initialization based on prediction information given to the branch

instruction.

15. (CURRENTLY AMENDED) A branch prediction method, comprising:  
performing a branch prediction in response to a branch instruction;  
updating a transition probability of the branch prediction according to whether a branch is actually made;  
detecting that a process is switched; and  
initializing branch prediction information when said detecting detects that the process is switched, and  
wherein said initializing includes fixedly performing initialization of the branch prediction information by determining an initialization value according to a comparison of a program counter value with a branch destination address and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN) ~~of the branch instruction to~~ set the branch prediction information to predetermined branch prediction information regardless of past branch prediction results, without depending on a particular process.

16. (CANCELLED)

17. (PREVIOUSLY PRESENTED) The information processing apparatus as claimed in claim 18, wherein the fourth part performs initialization based on prediction information given to the branch instruction.

18. (CURRENTLY AMENDED) An information processing apparatus comprising:  
a first part performing a branch prediction in response to a branch instruction;  
a second part updating a transition probability of the branch prediction according to whether a branch is actually made;  
a third part detecting that a process is switched; and  
a fourth part initializing branch prediction information when the third part detects that the process is switched, and  
wherein the fourth part fixedly performs initialization of the branch prediction information by determining an initialization value according to a comparison of a program counter value with a branch destination address of the branch instruction and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN), to set the branch

prediction information to predetermined branch prediction information regardless of past branch prediction results, without depending on a particular process.

19. (CURRENTLY AMENDED) A method of performing a branch prediction in response to a branch instruction, comprising:

detecting whether a process is switched; and

setting the branch prediction to predetermined branch prediction information regardless of past branch prediction results upon detecting that the process is switched, and

where the branch prediction information is initialized according to a branch destination address of the branch instruction, without depending on a particular process, based on a determination of an initialization value according to a comparison of a program counter value with the branch destination address and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN).

20. (CURRENTLY AMENDED) A method of performing a branch prediction in response to branch instructions, comprising:

storing branch prediction information based on past branch results in relation to the branch instructions;

detecting whether a process switch has occurred based on a program count address; and

setting a branch prediction in relation to one of the branch instructions to a predetermined branch regardless of the stored branch prediction information upon detecting that the process has switched in accordance with a determination of an initialization value based on a comparison of a program counter value with a branch destination address and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN).

21. (PREVIOUSLY PRESENTED) An arithmetic and logic unit, comprising:  
a first part performing a branch prediction in response to a branch instruction;  
a second part updating a transition probability of the branch prediction according to whether a branch is actually made;

a third part detecting that a process is switched; and

a fourth part initializing branch prediction information when the third part detects that the process has switched, and

wherein the fourth part performs initialization according to a branch destination of the branch instruction by determining an initialization value according to a comparison of a program counter value with a branch destination address and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN).

22. (PREVIOUSLY PRESENTED) A branch prediction method, comprising:  
performing a branch prediction in response to a branch instruction;  
updating a transition probability of the branch prediction according to whether a branch is actually made;  
detecting that a process is switched; and  
initializing branch prediction information when said detecting detects that the process has switched, and

wherein said initializing comprises performing initialization according to a branch destination of the branch instruction by determining an initialization value according to a comparison of a program counter value with a branch destination address and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN).

23. (PREVIOUSLY PRESENTED) An information processing apparatus, comprising:  
a first part performing a branch prediction in response to a branch instruction;  
a second part updating a transition probability of the branch prediction according to whether a branch is actually made;  
a third part detecting that a process is switched; and  
a fourth part initializing branch prediction information when the third part detects that the process is switched, and

wherein the fourth part performs initialization according to a branch destination of the branch instruction by determining an initialization value according to a comparison of a program counter value with a branch destination address and a determination of whether a branch prediction direction is backward taken (BT) or forward not taken (FN).

24. (NEW) A method of performing a branch prediction in response to a branch instruction, comprising:

setting the branch prediction to branch prediction information indicative of a branch prediction direction upon switch from a first process to a second process;

initializing the branch prediction information based on a comparison of a program counter value with a branch destination address; and

selectively outputting branch prediction direction information backward taken (BT) and branch direction information forward not taken (FN) based on the comparison such that when the branch destination address is smaller than the program counter value the branch prediction direction information backward taken (BT) is output.